

1 Complexite

2 LAB

3 Texture

4 Filtrage

5 Localisation

6 Sommation des reponses

7 Seuillage

8 Opérations topologiques

9 Contours

10 Annexe

11 Code

Complexite

LAB

Texture

Filtrage

Localisation

Sommation
des reponses

Seuillage

Opérations
topologiques

Contours

Annexe

Code

En théorie :

- Opérations vectorielles : GPU. $\nu(n)$
- Autres : CPU.

En pratique :

- Opération vectorielles : numpy, boucles sous-jacentes en C.
- Autres : python

Objectif : Vectoriser au maximum les opérations.

LAB

Complexité

LAB

Texture

Filtrage

Localisation

Sommation
des réponses

Seuillage

Opérations
topologiques

Contours

Annexe

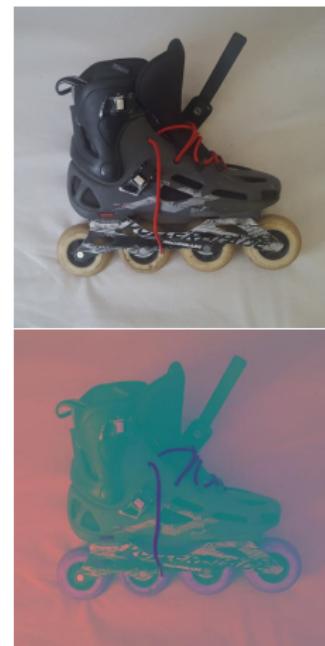
Code

$$L = \frac{R + G + B}{3}$$

$$A = \frac{G - R + 255}{2}$$

$$B = \frac{G - B + 255}{2}$$

Complexité : $\theta(\nu(h \cdot l))$



Transformée de Fourier

Complexité

LAB

Texture

Filtrage

Localisation

Sommation
des réponses

Seuillage

Opérations
topologiques

Contours

Annexe

Code

Définition :

- Continue : $F(f)(s) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-2i\pi sx} \cdot f(x) dx$
- Discrete : $FD(u)(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{-2i\pi \frac{kn}{N}} \cdot u_n$

Propriétés

- Involutive : $F(F(f)) = f$ $DF(DF(u)) = u$
- Symétrie : u réelle $\implies DF(u)$ paire
- Convolution : $u \star v = DF(u) \cdot DF(v)$ (Au facteur près)

Complexité temporelle

- Sur un vecteur : $\theta(l \cdot \nu(l))$
- Sur une image : $\theta((h \cdot l)(\nu(h) + \nu(l)))$

Complexité spatiale : En place

Transformée de Fourier rapide I

Complexité

LAB

Texture

Filtrage

Localisation

Sommation
des réponses

Seuillage

Opérations
topologiques

Contours

Annexe

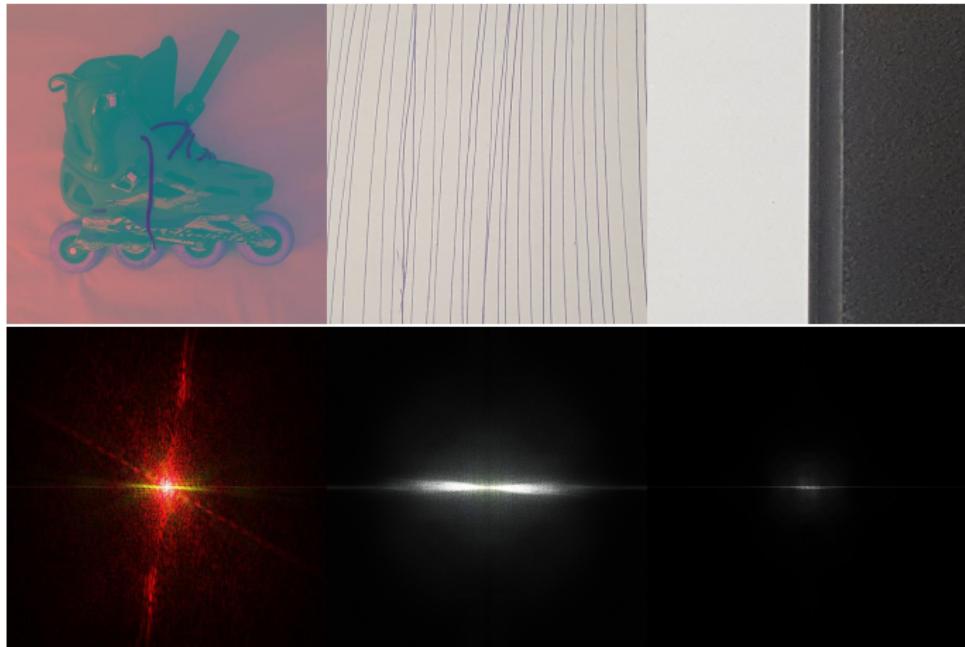
Code

Diviser pour régner
Complexité temporelle

- Sur un vecteur : $\theta(\log_2(l)\nu(l))$
- Sur une image : $\theta(\log_2(h)\nu(l \cdot h) + \log_2(l)\nu(h \cdot l))$

Exemples

Complexite
LAB
Texture
Filtrage
Localisation
Sommation
des reponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code



Filtres de convolution

Complexite

LAB

Texture

Filtrage

Localisation

Sommation
des reponses

Seuillage

Opérations
topologiques

Contours

Annexe

Code

Gaussienne d'orientation et
d'écart type variable

$$g(x) = e^{\frac{-x^2}{\sigma^2}}$$

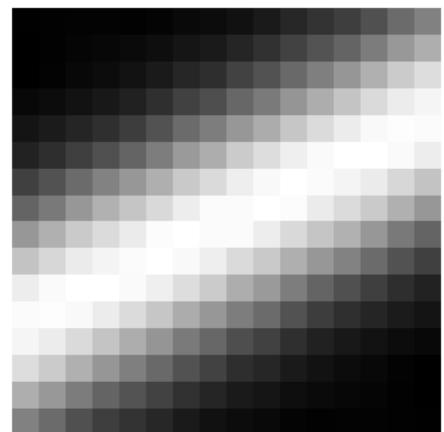


Figure – taille : 16, θ : $\frac{\pi}{6}$, σ : 4

Exemples

Complexite
LAB
Texture
Filtrage
Localisation
Sommation
des reponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code



Figure – tailled : 16, θ : $\frac{\pi}{2}$, σ : 4

Procédé

Complexité

LAB

Texture

Filtrage

Localisation

Sommation
des réponses

Seuillage

Opérations
topologiques

Contours

Annexe

Code

- Multiplication du voisinage de chaque pixel par le filter (centré sur le pixel)
- Sommation de toutes les valeurs

En pratique :

- On "décale" l'image de quelques pixels pour chaque coordonnée du filtre
- On multiplie chaque image décalée par le coefficient du filtre correspondant
- On ajoute les images ainsi obtenues

Complexité :

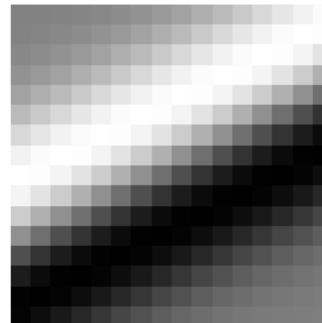
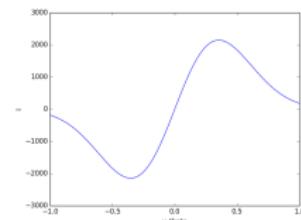
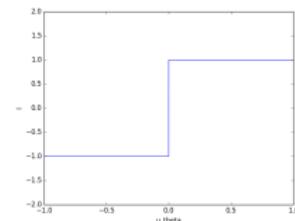
- Spatiale : En place
- Temporelle : $\theta(t^2 \cdot \nu(l \cdot h))$

Filtre de Canny

Complexite
LAB
Texture
Filtrage
Localisation
Sommation
des reponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code

Filtre optimal pour les aretes
en "pas" Dérivée d'une
gaussienne :

$$h(x) = x \cdot e^{-\frac{x^2}{\sigma^2}}$$



Exemples

Complexite
LAB
Texture
Filtrage
Localisation
Sommation
des reponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code

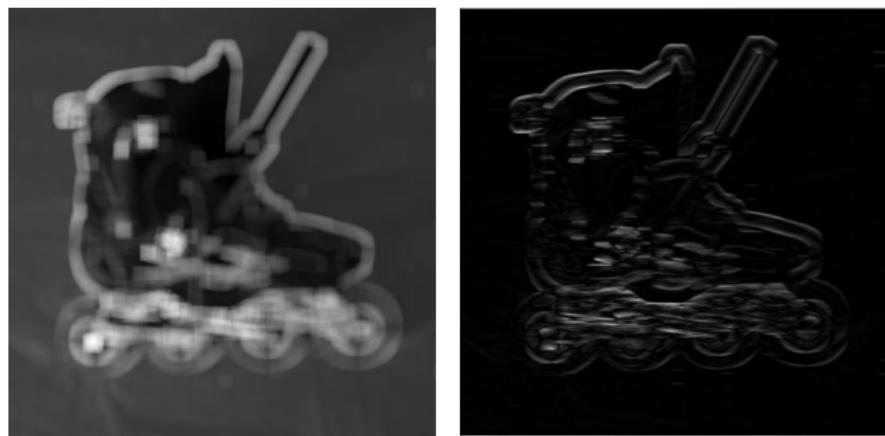


Figure – $\theta : \frac{\pi}{2}, \sigma : 1$

Complexité Temporelle

- Projection : $\theta(\nu(t^2))$
- Approximation parabolique :
 - Calcul des sommes : $\theta(t^2 \cdot \nu(h \cdot l))$
 - Résolution des systèmes : $\theta(\nu(h \cdot l))$
- Localisation : $\theta(\nu(h \cdot l))$
- Total temporel : $\theta(\nu(h \cdot l))$

Complexité spatiale :

$$\theta(t^2 \cdot h \cdot l)$$

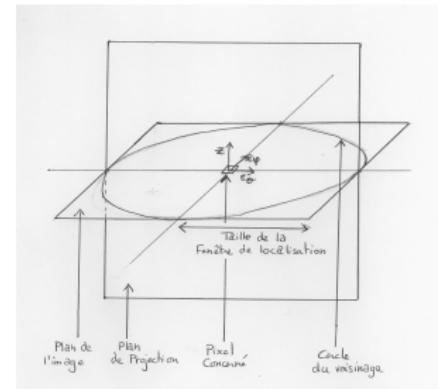


Figure – Projection

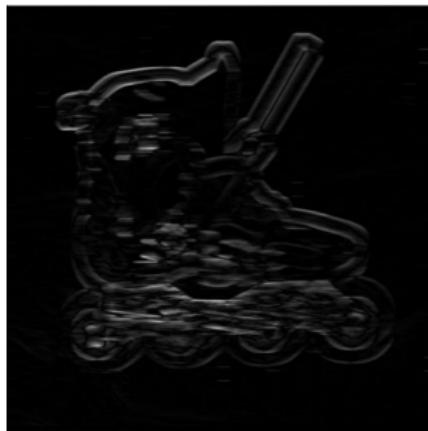
Approximation parabolique :

$$ax^2 + bx + c = 0$$

$$\text{Localisation : } \hat{f}(x) = c^+ \cdot \frac{2a^-}{|b|}$$

Exemple

Complexite
LAB
Texture
Filtrage
Localisation
Sommation
des reponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code



30 composantes :

- 3 couleurs + 3 couleurs * 4 directions = 15 composantes avant filtrage
- pour chaque composante, filtrage par deux filtres de Canny d'écart type 1 et 2 = 30 composantes

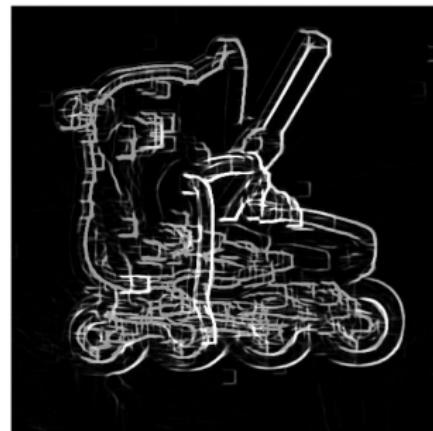
Choix des coefficients de sommation ?

Approche optimale : apprentissage supervisé

Ici : choix empirique

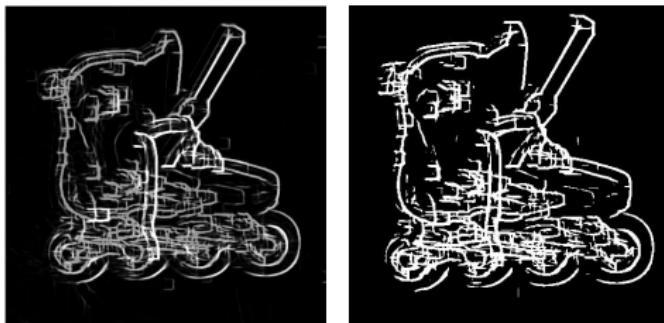
Exemple

Complexite
LAB
Texture
Filtrage
Localisation
Sommation
des reponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code



Complexite
LAB
Texture
Filtrage
Localisation
Sommation
des reponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code

Conversion en image binaire par seuillage :
1 si $I > \text{seuil}$
0 sinon



Définition

Complexité
LAB
Texture
Filtrage
Localisation
Sommation
des réponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code

Même procédé de filtrage mais sur des images binaires

Erosion

0	1	0
1	1	1
0	1	0

Dilatation

0	0	0
1	1	1
0	0	0

Fermeture

Dilatation +
Erosion

0	0	0	0	0
0	1	0	1	0
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1

0	0	0	0	0	0
0	1	1	1	0	0
0	0	0	0	1	1
1	1	1	1	1	1
1	1	1	1	1	1

0	0	0	0	0
0	0	1	1	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	1

0	0	0	0	0
0	0	0	0	0
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0

Exemples

Complexité

LAB

Texture

Filtrage

Localisation

Sommation
des réponses

Seuillage

Opérations
topologiques

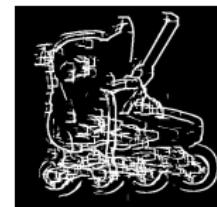
Contours

Annexe

Code

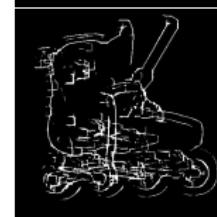
Erosion par :

0	1	0
1	1	1
0	1	0



Fermeture par :

0	0	0	0	0
0	0	0	0	0
1	1	1	1	1
0	0	0	0	0
0	0	0	0	0



Auquel on applique des rotations dans 8 directions différentes

Parcours du contour

Complexité

LAB

Texture

Filtrage

Localisation

Sommation
des réponses

Seuillage

Opérations
topologiques

Contours

Annexe

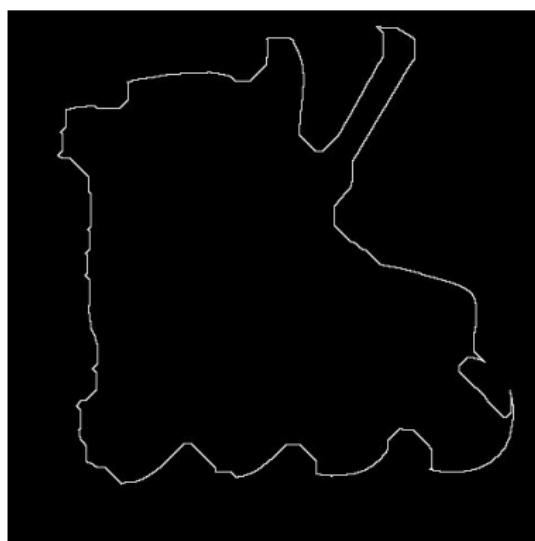
Code

Parcours dans le sens trigonométrique

Départ du pixel le plus en haut

Direction initiale : Bas

Priorités de déplacement : Gauche - Bas - Droite - Haut



Approximation affine

Complexite
LAB
Texture
Filtrage
Localisation
Sommation
des reponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code

Initialisation : Point le plus haut - Point le plus bas
Sur chaque segment, successivement :

- Calcul de D distance max du contour au segment
- Si $D >$ seuil, on ajoute le point le plus éloigné
- Lorsque pour tout segment $D <$ seuil, on sort de la boucle

Transformée de Fourier rapide I

Complexité
LAB
Texture
Filtrage
Localisation
Sommation
des réponses
Seuillage
Opérations
topologiques
Contours
Annexe
Code

Notation :

- $N = 2p$
- $n = n_1 + pn_2 \quad n_1 \in (0, p - 1) \quad n_2 \in 0, 1$
- $k = 2k_1 + k_2 \quad k_1 \in (0, p - 1) \quad k_2 \in 0, 1$
- $a = e^{-\frac{2i\pi}{N}}$

Alors :

$$F(u)(k) = \sum_{n=0}^{N-1} u(n) \cdot a^{kn} \quad (1)$$

$$F(u)(2k_1 + k_2) = \sum_{n_1=0}^1 \sum_{n_2=0}^{p-1} u(n_1 + pn_2) \cdot a^{k \cdot (n_1 + pn_2)} \quad (2)$$

$$= \sum_{n_1=0}^1 \sum_{n_2=0}^{p-1} u(n_1 + pn_2) \cdot a^{kn_1} \cdot a^{kp n_2} \quad (3)$$

Transformée de Fourier rapide II

Complexité

LAB

Texture

Filtrage

Localisation

Sommation
des réponses

Seuillage

Opérations
topologiques

Contours

Annexe

Code

De plus : $a^{kp n_2} = a^{2p \cdot k_1 n_2} \cdot a^{k_2 n_1 p} = a^{k_2 n_1 p}$

Ainsi : $F(u)(2k_1 + k_2) = \sum_{n_1=0}^1 a^{kn_1} \cdot \sum_{n_2=0}^{p-1} u(n_1 + pn_2) \cdot a^{k_2 n_1 p}$

Posons : $v(n_1, k_2) = \sum_{n_2=0}^{p-1} u(n_1 + pn_2) \cdot a^{k_2 n_1 p}$

Finalement : $F(u)(2k_1 + k_2) = v(0, k_2) + a^k \cdot v(1, k_2)$

Le calcul de v est celui d'une transformée de Fourier discrète.

On applique récursivement la décomposition